

## TITLE OF THE INVENTION

### SYNDICATION METHODOLOGY TO DYNAMICALLY PLACE DIGITAL ASSETS ON NON-RELATED WEB SITES

#### CROSS-REFERENCE TO RELATED APPLICATIONS

5           This application claims the benefit of U.S. Provisional Application No. 60/223,394  
filed August 7, 2000, entitled "SYNDICATION METHODOLOGY TO DYNAMICALLY  
PLACE DIGITAL ASSETS ON NON-RELATED WEB SITES."

#### COPYRIGHT NOTICE AND AUTHORIZATION

10           Portions of the documentation in this patent document contain material that is  
subject to copyright protection. The copyright owner has no objection to the facsimile  
reproduction by anyone of the patent document or the patent disclosure as it appears in the  
Patent and Trademark Office file or records, but otherwise reserves all copyright rights  
whatsoever.

#### BACKGROUND OF THE INVENTION

15           The present invention is directed to the process of Active Data Syndication and its  
use in web development and content management. More specifically, the present invention  
establishes the mechanism, the framework, the interfaces and the ancillary methodology to  
allow multiple content creators to offer for targeted syndication or network distribution and  
retrieval of digital assets across the web and to have those assets present within the constraints  
20   of the receiving web site, and to automate that process, where warranted.

          Conventional web sites consist of a wide variety of articles and information  
compiled and entered independently by a webmaster. Tools abound to facilitate the production  
of such web sites, from textual HTML editors to visual page designers like Adobe GoLive.  
Programs like LinkBOT exist to validate links among pages. However, the conventionality of  
25   this form of web site generation becomes merely a shell for the advent of the more advanced,  
dynamic and interactive web site.

Site management tools allow for the collaborative efforts of site creation, but are significantly limited in their ability to share information and digital assets across the boundaries of web sites.

Syndication or distribution of digital assets across the web heretofore involved the direct replication of those assets to a database server under the control of the receiving web site manager, and away from the control of the supplier of the digital asset. Examples of implementations where content replication is the primary form of asset syndication include those from Vignette, Kinecta, Interwoven, ICE, and ArcadiaOne.

Accordingly, there is a need for a system which (1) unifies the way digital assets, in any form, are shared from one site to another, (2) unifies the distribution of those digital assets across multiple platforms, (3) allows the owners of digital asset a management tool for tracking the business relationships surrounding the use of the digital assets, (4) establishes mechanisms, interfaces and methodology for the secure flow of information through information distribution networks, (5) facilitates the webmaster to have updated digital content present on the web site, and (6) automates the update of web content. The present invention fulfills these needs.

Prior to the present invention, it was not possible for web sites governed by any number of normal content management tools to proactively share information with independent and unrelated web sites. The lightweight and transportable tool of the present invention allows for a web site to be programmed to receive digital asset information in a multiplicity of formats, and to display those assets within the style and constraints of the receiving web site. The web sites need not be collocated, nor do they need to be created using the same tools. They simply need to render HTML in any web browser, capable of interpreting Java and JavaScript. The power of this tool is such that it can enhance, if not revolutionize, the bi-directional communications infrastructure using the web. These managed information distribution networks do for the Internet what faxes and overnight mail did for corporate correspondence two decades ago.

## SUMMARY OF THE INVENTION

The present invention provides a scheme of obtaining selected content for a web page, wherein the selected content itself is not initially part of the web page. The web page includes script, such as JavaScript, associated with the selected content. The scheme operates as follows:

1. A web browser requests a web page that includes script associated with the selected content. The selected content may be only a portion of the web page. The selected content may be a digital asset or an executable file. In the preferred embodiment, the web page is constructed using HTML, and the script is embedded therein.
2. The web browser interprets the script and formats a request to obtain the selected content from a remote site. The request includes a uniform resource identifier (URI) of the web page and a unique identifier of the selected content. The URI may be a URL.
3. A remote site, such as a web server, receives the request and authenticates whether the URI is authorized to receive the selected content. If so, then the remote site locates the selected content and sends the selected content to the web browser. The selected content may be stored in a content repository connected to the web server. If the URI is not authorized to receive the selected content, then the remote site sends a signal to the web browser that the selected content is not available, and the web browser assembles the web page without the selected content.
4. The web browser assembles the initially requested web page using the selected content obtained from the remote site.

The assembled web page may include one or more content sets from the syndicator, each having its own script for implementing the steps above.

In one preferred embodiment, the script includes a subscriber identifier and a content identifier which are both used to create the unique identifier of the selected content.

The present invention also provides a scheme for syndicating digital assets. A web page is constructed, and script, such as JavaScript, associated with at least one digital asset that is desired to be part of the fully rendered web page is inserted into the web page. The script,

when executed by a browser, performs modest authentication regarding the URI and requests the content of the digital asset from content repository. The request includes a uniform resource identifier (URI) of the web page and a unique identifier of the selected content. In one preferred embodiment, the script includes a subscriber identifier and a content identifier, which, together, create the unique identifier of the selected content. The selected content may be an executable file.

## BRIEF DESCRIPTION OF THE DRAWINGS

The following detailed description of preferred embodiments of the present invention would be better understood when read in conjunction with the appended drawings.

For the purpose of illustrating the present invention, the drawings show embodiments of the present invention which are presently preferred. However, the present invention is not limited to the precise arrangements and instrumentalities shown. In the drawings:

Fig. 1 is a schematic block diagram that provides an overview of one preferred embodiment of the asset syndication scheme of the present invention;

Fig. 2 is a database schema for one preferred embodiment of the present invention;

Fig. 3 is an authentication schema for one preferred embodiment of the present invention;

Figs. 4-13B are user interface displays (administrative screen shots) for one preferred embodiment of the present invention;

Fig. 14 is a database schema for the second preferred embodiment of the present invention;

Figs. 15A and 15B, taken together, are overall schemas for the second preferred embodiment of the present invention;

Fig. 16 is a schematic block diagram of the second preferred embodiment of the present invention;

Figs. 17-19 are JavaScript source code snippets for implementing a web application embodiment of the present invention; and

Fig. 20 is a database schema for the web application embodiment of the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

Certain terminology is used herein for convenience only and is not to be taken as a limitation on the present invention. In the drawings, the same reference letters are employed for designating the same elements throughout the several figures.

### 5 I. DEFINITIONS

Content: Any textual, visual, and audio materials or the combination thereof, including animated images, video clips, executable files, or digital assets targeted for presentation.

Content Repository: a database of file structure which contains the syndicable asset.

Syndicated Asset: Any content which is available to an authenticated receiver.

10 Authenticated Receiver: A receiver of content that is authorized to retrieve and present that content.

Subscriber: Synonym to Authenticated Receiver.

Domain: The name which appears in the URL between the "www" and the end of the three letter extension (e.g., .com, .net, .org). Example: www.regiononline.com, the domain is  
15 regiononline.com.

### II. OVERVIEW OF PRESENT INVENTION

Referring to Fig. 1, the following steps are performed:

1. A request is made from a web appliance to a URL, which contains a web page that has embedded the syndication code.
- 20 2. The web page, while rendering other data present on the page, such as navigation, headers and footers, and other assets not related to the syndicated content, returns JavaScript which tests the appliance for the site URL requested. The JavaScript effectively functions as a dynamic content retrieval agent.
- 25 3. This information is then used to authenticate the content presence on the site as valid, and then to retrieve from the syndication database those assets for display, and render those syndicated assets through the appliance.

1 The present invention introduces the notion of web real estate and the apportionment  
of content sections to a particular web page. A visually unified site is substantially divided into  
sub-sites, or content sections, that are independently maintained by section managers or through  
syndicated content offerings. As a direct and targeted information management delivery tool,  
5 the present invention is useful to the large corporation, a community site, or any site which  
enables the sharing of targeted information. Any content or information rich site will find this  
present invention very helpful.

10 The present invention may be used to complement web pages created in MS  
Frontpage, or any number of content management tools. The present invention does not  
perform visual layout, and may be considered a post-design tool.

15 The present invention may be used with any number of programming languages  
such as Cold Fusion, ASP, C++, Java, Visual Basic or Perl. The present invention may serve as  
an extension to the web page, or an add-on component to any number of content management  
tools.

20 The present invention may be used on any site that renders HTML, and is intended  
to be programming language neutral, through the use of JavaScript, servlets, and Java features.

### III. DETAILED DESCRIPTION OF ONE PREFERRED EMBODIMENT

25 In one preferred embodiment of the present invention as described herein, the user  
communicates through a browser with the necessary web site via an electronic network, such as  
the Internet. However, the scope of the invention includes other types of user interfaces and  
electronic networks that are capable of performing the desired functions.

The present invention is described in the context of a commercially available  
software product called Active Data Syndicator™, available from Active Data Exchange, Inc.,  
Bethlehem, Pennsylvania.

#### A. DETAILED EXPLANATION OF FIGURES AND APPENDICES

Fig. 2 is a self-explanatory database schema for one preferred embodiment of the  
present invention, and Fig. 3 is a self-explanatory authentication schema for one preferred  
embodiment of the present invention.

Fig. 4 is an administrative entry screen for beginning the process.

Fig. 5 is a user interface display that allows for the selection from an existing syndication or the creation of a new syndication.

Referring to Fig. 6, if the administrator chooses the “go get it” button from the Fig. 5 display, then a listing of the syndication offers are presented. The “clickable” first field  
5 allows for the modification of that offering.

Figs. 7 and 8 are user interface displays a for a “New Setup.” The administrator completes the fields shown in these displays.

Fig. 9 is a user interface display that allows for the viewing of the necessary syndication code for placement into the HTML of the subscriber/receiver web site.

Fig. 10A shows the actual code for placement on the subscriber/receiver HTML page for the fictitious client Attorney at Law. This code allows for the one time insertion of the code for presentation of information, news, events, or other digital assets on an ongoing basis.  
10

Fig. 10B shows the same code, modified to present the latest asset, as well as the creation of several hypertext links for viewable archives of older digital assets. In this case,  
15 there is the ability to present for viewing up to 10 viewable archives from the database.

Figs. 11 and 12 show additional administrative functions for the deletion of a subscriber.

Figs. 13A and 13B show the list of subscribers/receivers for overall administrative purposes.  
20

Appendix A is the syndication source code for the embodiment of the present invention shown in Figs. 2-13B.

Appendix B is the servlet package source code for the embodiment of the present invention shown in Figs. 2-13B.

## B. INHERENT SECURITY

All web sites do not use the same syndication code. Subtle differences in the code are “keyed” to the receiving domain. It is through this “key” that the integrity of the business relationship and the placement of the digital asset are preserved. The process includes a security module that looks at the browser URL of the appliance to determine which URL (domain) it is calling. If the domain does not match an existing client URL (coupled with the  
30 content authorized for use), a message is returned to the appliance, which states that the content is unavailable.

### C. WEB SITE SETUP FOR RECEIVING SYNDICATION OFFERINGS

To set up a web site page for syndication, a few parameters must be understood and defined:

1. The receiving URL, as authentication or validation of the receiver/subscriber is performed to the domain level.
2. The asset section to be syndicated. This refers to a content section in a content management system. One instant example would be to syndicate "What's News" from a company or organization. This section of the site is a hypothetical region where all information regarding press releases and latest organizational information is to be found. In this installation, the site only needs to be set up with the HTML snippet one time. Content changes occur dynamically.
3. The number of displayed archives the receiver would like to be able to view. This is a title listing of previously syndicated content for this section.
4. The database location of the digital asset.
5. The organizational name (for administrative purposes).
6. The start and end time for the syndication. This is important if the syndicated asset is time dependant or whether or not the subscription to the syndicated digital asset is on a monthly or other time dependent basis.

Through these parameters, metrics can be derived which display, from the syndication-offered standpoint, several management reports useful in understanding the offerings and their use. Metrics include:

1. Number of times a digital asset is accessed by a subscriber (useful in fee per use models as well as for tracking relative worth of the asset).
2. Places where the digital asset can be viewed/authorized domains.
3. Listing of the offerings available for release through the syndication model.
4. Through an oblique methodology, a measure of the physical overhead of the device housing that content.

### D. DATABASE INTERRELATIONSHIPS

As discussed above, the present invention is platform independent and program neutral. It functions equally as well on operating systems written in Microsoft, as it does from



that of Red Hat Linux. Furthermore, the database from which it pulls its content may either be MS SQL, Informix, Sybase or Oracle, as it uses command phrases which are non-vendor specific (ANSI SQL). The hardware, however, must be of sufficient strength to power databases that conceivably will receive hundreds of thousands of requests per second. Fig. 2,  
5 described above, illustrates one preferred database schema for the present invention.

#### E. ARCHITECTURAL OVERVIEW

The implementation of this business process and its underlying software is hardware independent, providing that the following applications/services are available: An application server running a servlet engine such as Allaire's JRUN or TomCat, and a web server such as  
10 Internet Information Server (IIS) from Microsoft or an Apache web server should the operating system of the server be RedHat Linux or Unix. The database server should have a strong relational database such as MS SQL, Oracle, or Sybase, overtop of the appropriate operating system.

The location of the servers used in the present invention can be anywhere within the  
15 infrastructure of the existing client network. One suitable scheme is to locate the servers behind a redundant firewall.

The invention environment provides for all system hardware requisite to the invention work. It includes an application/web server that controls the look of web pages, serves content for the web pages, and provides the environment for the administration of those  
20 tasks. The standard operating system for this invention is Linux. However, a client may choose Windows NT or UNIX as an alternate operating system (OS). Coupled with the OS, a web server needs to be chosen. In the example of Red Hat Linux, one suitable choice would be Apache. However, a Windows NT environment would most likely use IIS. Since the present invention is a Java-based application, a servlet engine needs to be incorporated into the server  
25 operation. One suitable configuration to support the invention includes the following elements:

##### Operating System (Application Server)

Windows NT 4.0 with Service Pack 6A

Red Hat Linux 6.2

##### Databases (Database Server)

30 Sybase 11.02 [Windows NT, Linux]

Oracle 8i Server 8.1.5 [Windows NT, Solaris, Linux]

Microsoft SQL Server 6.5 with Service Pack 6A [Windows NT Only]

Microsoft SQL Server 7.0 [Windows NT Only]

#### JDBC Drivers

5           Microsoft SQL Server 6.5, 7.0: SPRINTA JDBC 2.0 Driver version 2000 [Type 4]

Oracle 8.0.5.0. Production release JDBC [Windows NT, Solaris, Linux]

#### HTTP Servers (Web Server Types) Operating JRUN 2.3.3 or greater

Netscape Enterprise Server 3.0, 3.6 [Windows NT, Solaris, Linux, AIX]

Microsoft Internet Information Server Version 4.0 [Windows NT Only]

10          Apache and Stronghold Web Server 2.4.2 [NT, AIX, Solaris, Linux]

#### Java Development Kit (JDK)

Recommendation: Sun Compliant JDK minimum version 1.2.2

#### Example Configurations:

1.           Linux Application Server with NT OS running the Database server  
15           Application/Web Server: Red Hat Linux version 6.2 with an Apache web server.  
Database Server: Microsoft NT 4.0 with Service Package 6A and MS SQL version 7  
as the database.
2.           Linux Application Server with Linux OS running the Database server  
Application/Web Server: Red Hat Linux version 6.2 with an Apache web server.  
20           Database Server: Red Hat Linux version 6.2 with Oracle 8i as the database.
3.           NT Application Server with NT OS running the Database server  
Application/Web Server: Microsoft NT 4.0 with Service Package 6A and IIS  
version 4 with full options as the Web Server.  
25           Database Server: Microsoft NT 4.0 with Service Package 6A and MS SQL version 7  
as the database.

The Database Server becomes more critical as database size increases over time. It is feasible for an installation to use a different OS for the application server than that of the database server. Different database engines serve data at different rates (some faster than others) and some networks operate with corporate standards which mandate the use of certain configurations, to standardize application environments for ease of maintenance. For those

30

reasons, the publishing database was developed platform independent. It is functionally transparent whether or not it is installed over Oracle, Sybase, MS SQL or Informix.

The present invention may be installed as part of a related product from Active Data Exchange, Inc., called Active Data Publisher™/Web Server, or it may be a functionally independent device. The setup of the environment of the present invention is the same as the Application/Web server environment.

#### F. EXAMPLES OF SYNDICATION USE AND APPLICATION

##### Case One: Large organization with multiple web sites (Public and Private)

A large organization has multiple web sites to manage, both public (visible to all) and private (internal and departmental in nature). The sites are repositories for a wide variety of information specific to their department or corporate division. There are, however, many assets which are frequently reproduced and shared among those sites. If a directive from Senior Management needs to be presented on each of the sites, conventional content management tools would require the HTML editors to re-post the directive on each and every site. This repetitive process creates organizational inefficiencies.

Using the present invention, an HTML snippet is placed on the page in the place where information bulletins will be seen. (See Figs. 10A and 10B which illustrate an HTML snippet.) Every time a new information bulletin is released, the web page is automatically updated, without technology intervention. Furthermore, the bulletin is created in the native environment frequently used by the author or their assistant, and posted once through an intuitive, easy to use interface module which converts the document to HTML, enters the data into the syndication database, and prepares it for view on the site.

##### Case Two: Supply Chain Information Management

A manufacturer manages information streams from raw materials suppliers (upstream providers) to distributors (end users). The management objective is to assure that the end user receives the completed goods at a fair price in a timely manner, and that the raw materials suppliers notify the manufacturer of delays in shipments, which, in turn, affect product availability. If one believes that the main differentiation between suppliers of similar materials is their ability to enhance communications with clients and vendors, and that issue singularly is what is going to set apart one from another, then the implementation of the tools

provided by the present invention is the linchpin in corporate success. In this example, it is helpful from a supply management role to understand the needs of the client and the ability for the vendor to supply product to create the materials for the client. Subscribing to the notion of apportioned web real estate, and having particular areas designated for information relevant to those in the supply chain, a new communications modality is created called a Digital Information Network that is linking the supplier and the end user in such a fashion as to be able to enhance the decision process, increase productivity, and enhance the digital economy.

#### IV. DETAILED DESCRIPTION OF A SECOND PREFERRED EMBODIMENT

The second preferred embodiment may be used with Active Data Syndicator v.3.

##### A. DETAILED EXPLANATION OF FIGURES AND APPENDIX

Fig. 14 is a self-explanatory database schema for the second preferred embodiment.

Fig. 14 shows only the portion of the database schema that relates to the present invention.

Figs. 15A and 15B, taken together, are self-explanatory overall schema for the second preferred embodiment.

Fig. 16 is a schematic block diagram of the second preferred embodiment.

Appendix C is the combined syndication and servlet package source code for the embodiment of the present invention shown in Figs. 14-16.

##### B. IMPLEMENTATION OF SECOND EMBODIMENT

Fig. 16 shows an overview of the second preferred embodiment. The basic elements include web pages located at a plurality of different URLs, a viewing browser, an application web server that hosts the syndication product, and a content repository. Each of these elements may be interconnected by any suitable communication medium, such as the Internet. The process operates as follows:

1. A user at the viewing browser requests a web page from a particular web site. The requested web page contains HTML elements, as well as at least one snippet of JavaScript associated with the syndicated digital asset. Alternatively, the requested web page does not have to reside on a web site. The requested web page may also be a simple HTML file stored on the hard drive of a user's local computer in cases where an authenticating URL is not required.
2. The snippet of JavaScript is received by the browser.

3. When the JavaScript is received, it is not immediately used by the browser on the part of the web page on which it resides. Instead, the “src” attribute of the JavaScript tag is used to make a call to a Java servlet. More specifically, the JavaScript forms an HTTP (or HTTPS) request that includes a URI (e.g., a URL) of the requested web page as obtained from the browser, and a unique identifier contained within the JavaScript. In one preferred embodiment, the unique identifier is a combination of a subscriber identifier (sub ID) and a content identifier (content ID). In effect, the HTTP request is saying that URI [xyz] is requesting content [123], and is asking if it is okay to deliver it. The HTTP request is sent to an application web server that is designated by an address located within the JavaScript.
4. At the application web server, the syndication product receives the URI and the unique identifier,
5. The syndication product then looks in an authentication table to determine if there is a matching URI and unique identifier. If so, then the unique identifier is parsed to obtain the sub ID and content ID. The sub ID is used to check the current account status of the subscriber, and the content ID is used to locate the content in the content repository.
6. Assuming that a match is found in the authentication table, the subscriber ID is properly authorized, and the content is located in the content repository, then the content is retrieved from the content repository and sent by the syndication product to the browser for insertion at the appropriate location during the rendering of the web page. To facilitate this process, the syndication product contains content filtering and parsing methods (called “parseContent” and “swapStrings” in the example source code) which are used to prepare the content so that it can be rendered appropriately in the web page via a JavaScript “document.write” statement. If no match is found in the authentication table, and/or if the subscriber ID is not properly authorized, then a message is returned indicating that requested content cannot be received. Alternatively, no message is returned and the user merely does not receive the requested

content. If the content is a text article, the web page may have a blank portion where the requested content would have appeared. If the content is multimedia-oriented, such as an audio file, then such content is not experienced.

- 5           7. The syndication content manager updates its records to reflect the activity. The manager may track content retrievals and charge subscriber accounts (if any exist) for such content retrievals. The content manager may remove an entry from the authentication table based upon expiration dates, number of retrievals, or any other suitable factor.

10           The process described above preferably occurs seamlessly in near real-time. Thus, the user is not aware that content (which is typically only a portion of the web page, but could be the entire content of the web page) is being requested and delivered from a remote content repository during the rendering of the web page.

15           The term "web application" as used herein refers to dynamic HTML web site content which varies depending upon user input, includes one or more interactive forms, involves the use of a web server programming/scripting language (e.g., Java, Perl, Cold Fusion, Active Server Pages, etc.), and may also make use of a backend database server for data storage. Some common examples of web applications include guestbooks, forums and shopping carts. Web applications typically execute on the same server as the hosting web  
20           server. This arrangement can place significant strain on the web server, especially when a large number of users are simultaneously requesting service and/or many web sites and applications are running concurrently on the same server. (This situation often occurs with ISP's.) The present invention leverages the JavaScript capabilities of the user's browser to execute the web application at a remote server independent of the subscriber's hosting web server, thereby  
25           reducing the potential load on the subscriber's web server and greatly simplifying the process by which a web application can be incorporated into a subscribing web site. That is, simply include the JavaScript snippet of the application using the present invention's syndication methodology in the HTML on the subscriber's web site. No further programming is then needed on the subscriber's web server. The user's browser effectively invokes the remote web  
30           application that runs and makes the subscriber's web page dynamic.

The present invention may be used for flat (static) sites and flat sites having one or more dynamic sections, as well as for fully dynamic sites.

The present invention is further advantageous because the traffic for creating the web pages uses port 80 (for http traffic) and port 443 (for https traffic), and thereby can pass through most server firewalls.

### C. WEB APPLICATION EXAMPLE

Active Data Randomizer, available from Active Data Exchange, Inc., Bethlehem, Pennsylvania, is an example of a simple web application which incorporates the syndication methodology of the present invention. Randomizer produces two syndicated assets from two different JavaScript snippets.

Fig. 17 shows the first JavaScript snippet which renders in the browser as an administrator area (see Fig. 18) for the subscribing web site administrator to use in the configuration and entry of groups of HTML blurbs.

Fig. 19 shows the second JavaScript snippet which is generated from the administrative area and renders in the browser as a randomly selected HTML blurb from a group of HTML blurbs specified during the generation of the snippet.

Appendix D is a User Guide for this embodiment, and Appendix E shows sample source code for this embodiment.

The present invention may be implemented with any combination of hardware and software. If implemented as a computer-implemented apparatus, the present invention is implemented using means for performing all of the steps and functions described above.

The present invention can be included in an article of manufacture (e.g., one or more computer program products) having, for instance, computer useable media. The media has embodied therein, for instance, computer readable program code means for providing and facilitating the mechanisms of the present invention. The article of manufacture can be included as part of a computer system or sold separately.

It will be appreciated by those skilled in the art that changes could be made to the embodiments described above without departing from the broad inventive concept thereof. It is understood, therefore, that this invention is not limited to the particular embodiments disclosed,

but it is intended to cover modifications within the spirit and scope of the present invention as defined by the appended claims.

What is claimed is:



## CLAIMS

1. A method of obtaining selected content for a web page, wherein the selected content itself is not initially part of the web page, the web page including script associated with the selected content, the method comprising:

(a) a web browser requesting a web page that includes script associated with the selected content; and

(b) the web browser interpreting the script and formatting a request for obtaining the selected content from a remote site, the request including a uniform resource identifier (URI) of the web page and a unique identifier of the selected content.

2. A method of claim 1 further comprising:

(c) a remote site receiving the request and authenticating whether the URI is authorized to receive the selected content, and, if so, the remote site locating the selected content, and sending the selected content to the web browser; and

(d) the web browser assembling the initially requested web page using the selected content obtained from the remote site.

3. The method of claim 2 wherein the remote site is a web server, and the selected content is stored in a content repository connected to the web server.

4. The method of claim 2 wherein the selected content includes two or more different selected content, each selected content being used for different parts of the web page, wherein each selected content has its own script for implementing steps (b)-(d).

5. The method of claim 2 wherein in step (c), if the URI is not authorized to receive the selected content, the remote site sends a signal to the web browser that the selected content is not available, and the web browser assembles the web page without the selected content.

6. The method of claim 1 wherein the URI is a uniform resource locator (URL).

7. The method of claim 1 wherein the selected content is only a portion of the web page.

8. The method of claim 1 wherein the selected content is a digital asset.

9. The method of claim 1 wherein the selected content is an executable file.

10. The method of claim 1 wherein the script includes a subscriber identifier and a content identifier, and step (b) further comprises using the subscriber identifier and the content identifier to create the unique identifier of the selected content.

11. The method of claim 1 wherein the web page is constructed using HTML, and the script is embedded therein.

12. The method of claim 1 wherein the script is JavaScript.

13. A method of syndicating digital assets comprising:

(a) constructing a web page; and

(b) inserting into the web page script associated with at least one digital asset that is desired to be part of a fully rendered web page, wherein the script, when executed by a browser, requests the content of the digital asset from a remote site, the request including a uniform resource identifier (URI) of the web page and a unique identifier of the selected content.

14. A method of claim 13 wherein the script is JavaScript.

15. A method of claim 13 wherein the selected content is an executable file.

16. A method of claim 13 wherein the script includes a subscriber identifier and a content identifier, which, together, create the unique identifier of the selected content.

17. An article of manufacture for syndicating digital assets, the article of manufacture comprising a computer-readable medium holding computer-executable instructions for performing a method comprising:

(a) constructing a web page; and

(b) inserting into the web page script associated with at least one digital asset that is desired to be part of a fully rendered web page, wherein the script, when executed by a browser, requests the content of the digital asset from a remote site, the request including a uniform resource identifier (URI) of the web page and a unique identifier of the selected content.

18. The article of manufacture of claim 17 wherein the script is JavaScript.

19. The article of manufacture of claim 17 wherein the selected content is an executable file.

20. The article of manufacture of claim 17 wherein the script includes a subscriber identifier and a content identifier, which, together, create the unique identifier of the selected content.

21. An apparatus for syndicating digital assets comprising:

(a) means for constructing a web page; and

(b) means for inserting into the web page script associated with at least one digital asset that is desired to be part of a fully rendered web page, wherein the script, when executed by a browser, requests the content of the digital asset from a remote site, the request including a uniform resource identifier (URI) of the web page and a unique identifier of the selected content.

22. The apparatus of claim 21 wherein the script is JavaScript.

23. The apparatus of claim 21 wherein the selected content is an executable file.

24. The apparatus of claim 21 wherein the script includes a subscriber identifier and a content identifier, which, together, create the unique identifier of the selected content.

## ABSTRACT OF THE DISCLOSURE

Digital assets are syndicated by constructing a web page and inserting into the web page JavaScript associated with a digital asset that is desired to be part of a fully rendered web page. The content of the digital asset is not initially part of the web page. The script, when executed by a browser, requests the content of the digital asset from a remote site. The request  
5 includes a uniform resource identifier (URI) of the web page and a unique identifier of the selected content. The remote site receives the request and authenticates whether the URI is authorized to receive the selected content. If so, the remote site locates the selected content and sends the selected content to the web browser. The web browser assembles the initially requested web page using the selected content obtained from the remote site.